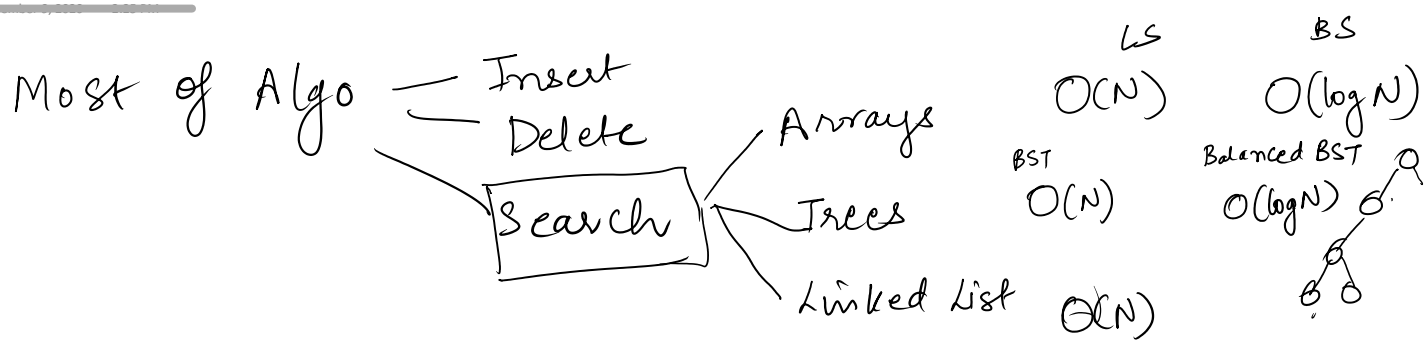


Task- Search(x) = comparing "x" with other elements

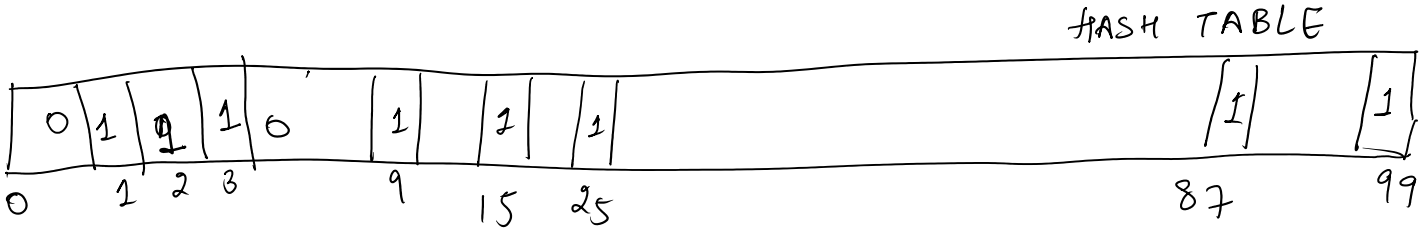
Q. // Can we Search(x) without any comparison?

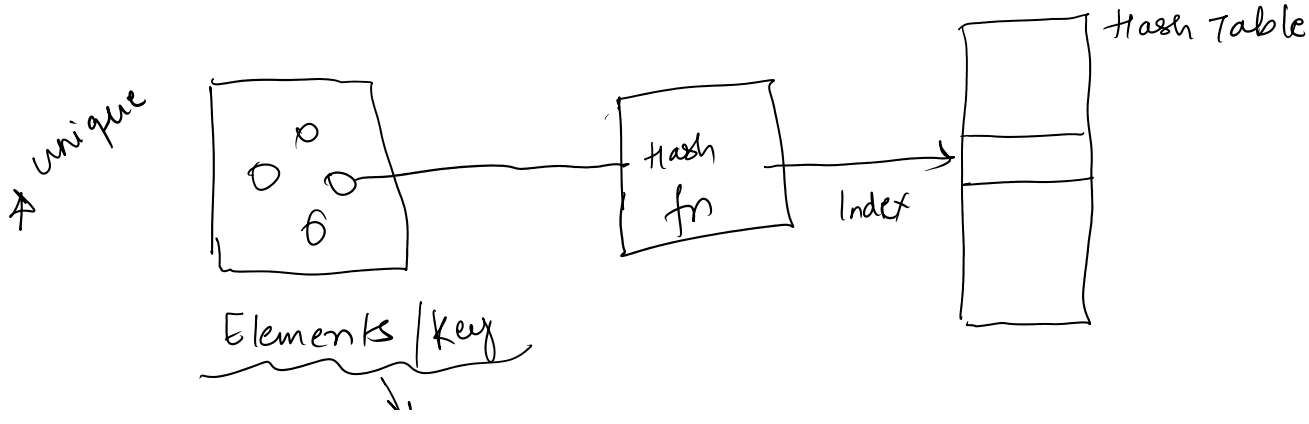
Locate/Search - O(K) → K is constant

HASHING

Eg. Array of 10 integers, 0 - 99, unique integers

[1, 3, 9, 15, 25, 87, 99, 2, 42, 50]





Elements / Key

We use values as the I/P or index of Hash function

KEY BASED ALGORITHMS → DIRECT ADDRESSING

Given a key value we can find the item in O(1)

- Hash function

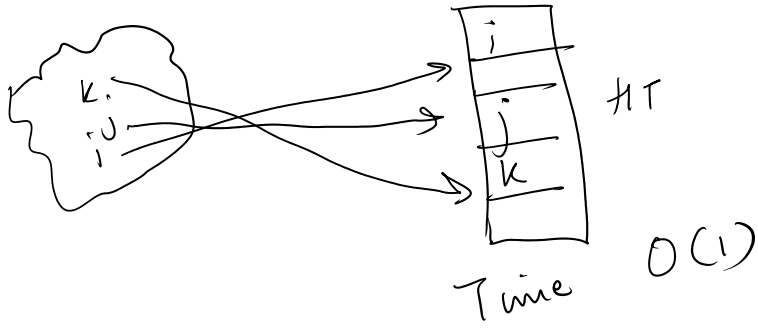
$y = h(x)$ — unique

y → O/P index (might not be unique)

x → I/P / key

operation that maps one large set of values into another set of values

Time vs Space complexity.



#g Student Ids – 10,000 students

Hash function – Last 4 digits → index of the HT

7213

HT - 10,000

⇒ multiple student ids pointing to the same slot in HT

↳ COLLISION

If hash functions points to the same index in the hash table

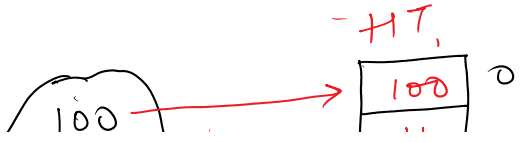
- Better hash functions
- Deal with collissions

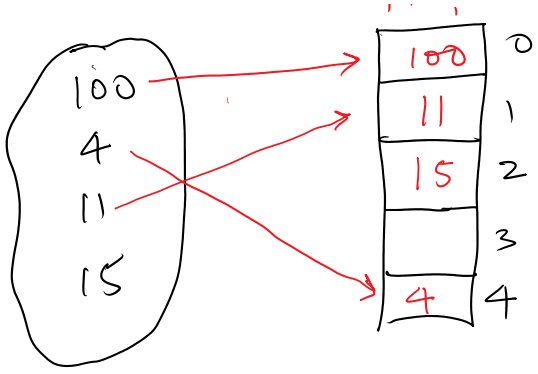
★ COLLISSION RESOLUTION

- Linear Probing
  - Open Addressing
- Quadratic Probing
- Pseudo-random probing
- Double Hashing

Linear Probing

Eg 100, 4, 11, 15 $h(x) = X \% 5$; size of HT = 5





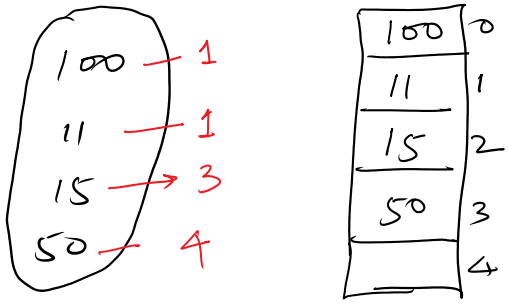
Many to one function — Collisions are inevitable

Find the index of the HT using hash function.

If that index is not empty, traverse the hash table linearly (go to the next index).

Keep on doing the above recursively, till an empty slot is found.

#q 100, 11, 15, 50 $h(x) = x \% 5$

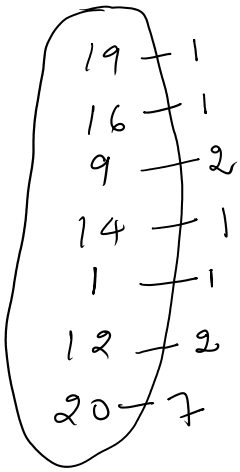


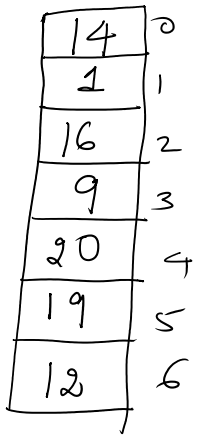
no. of probes

$= 1 + 1 + 3 + 4$

$= 9$

#q





$h(x) = x \% 7$

$LF = \frac{7}{7} = 1$

⇒ **LOAD FACTOR** = $\frac{\#\text{ of items stored in HT}}{\text{Total size of HT}}$

#q

$LF = \frac{3}{5}$

If HT is 2/3 full — spend most of the time doing collission resolution. For an average cluster size of 3, avg 2 probes

collissions tend to form clusters

↓

group of items

Clusters grow quickly — SNOWBALL EFFECT

⇒ HT is sparse — Linear Probing is faster

HIT

$$\frac{1 + \frac{1}{1-x}}{2}$$

Find an item

Knuth's formula

Avg. no. of probes

MISS

$$\frac{1 + \frac{1}{(1-x)^2}}{2}$$

Discover that the item does not exist

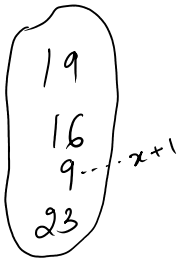
* Quadratic Probing

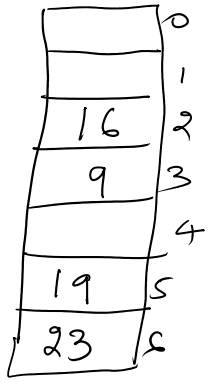
Probes = {1, 4, 9, 16, ... y}

$\Longrightarrow$ $y+1$, $y+2^2$, $y+3^2$, + - - - -

index from the hash fn

#g.





$h(x) = x \% 7$

for 9 $- h(x) = 2 = y$

collission

$y+1 = 3$

for 23 $- y = 2$

collission $y+1 = 3$

$y+2^2 = 6$

* Pseudo - Random Probing

{ 1, 5, 7, 6, ~~5~~, 8, 3, 4, 0, 2 }

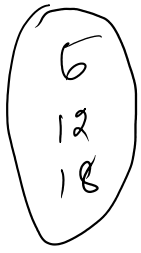
* Double Hashing — P(k)
— S(k)

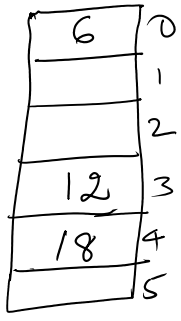
1st Probe P(k) + S(k)

2nd Probe P(k) + 2S(k)

3rd Probe P(k) + 3S(k)

#g 6, 12, 18 $h(x) = x \% 6$; $g(x) = (x \% 5) + 1$



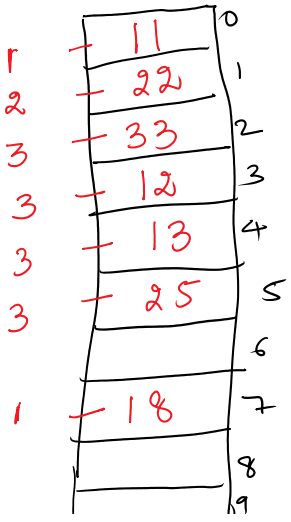


for 12 $= \overset{h(x)}{0} + \overset{g(x)}{3} = 3$

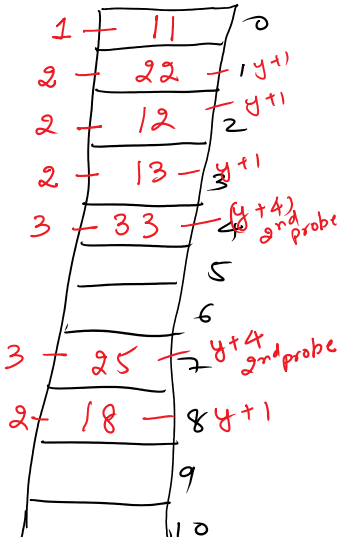
for 18 — $h(x) + g(x)$
$= 0 + 4 = 4$

#g 11, 22, 33, 12, 13, 25, 18 $h(x) = x \% 11$; $g(x) = (x \% 7) + 1$

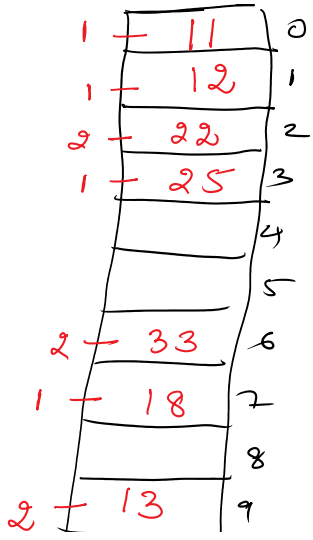
LP



QP



DH

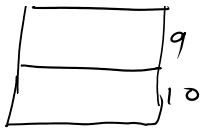


$22 = \overset{h(x)+g(x)}{0+2}$

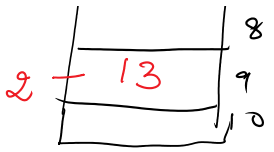
$33 = 0 + 6$

$13 = 2 + 7$

probes = 16



probes = 15



probes = 10

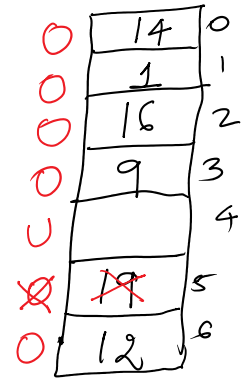
① Deletion from Hash Table — Mark / Pack

Every slot in HT — O / U / D

occupied — un-occupied — Deleted

⇒ Mark — O/U/D , If inserting U/D

Eg 19, 16, 9, 14, 1, 12 — $h(k) = k \% 7$



Delete 19
↓
remove it from HT
change/Mark — D

Find 12 — 12 % 7 = 5 — O → U

↳ Slot 5 - D → look for the next slot

So if slot 5 - U → 12 does not exist in HT

U → If nothing is stored in that slot

O → If there is a key value in that slot

D → If there was a key value previously stored in that slot, & now that value has been removed from HT

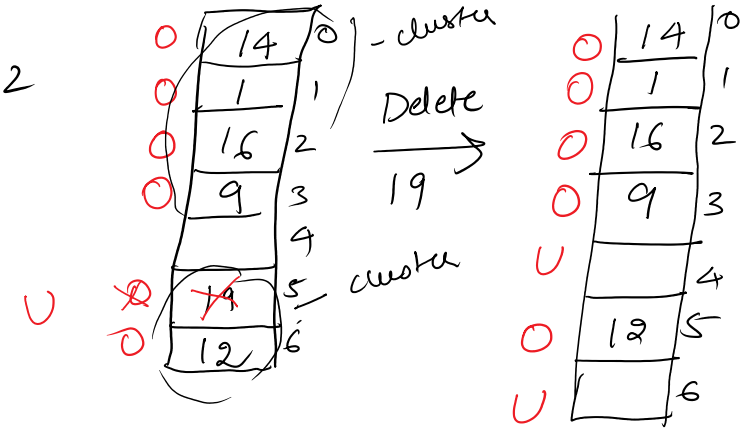
Insert '26' — index 5 — mark D

insert 26 in slot 5 & change mark O

⇒ Pack — Reordering

1. for each item in HT after the deleted item that is in the cluster, mark its slot as unoccupied and reinsert in the HT.

#g. 19, 16, 9, 14, 1, 12



Delete 19 from HT

change slot 5 mark O → U

Reinsert 12 — find the index for 12 in HT again

Insert '26' — in slot 6 because slot 5 is O

✱ CHAINING

Linked list

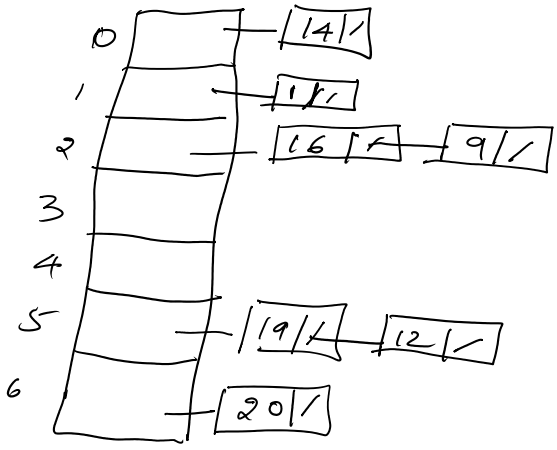
Data is stored outside of the table

#g 19, 16, 9, 14, 1, 12, 20
1 1 2 1 1 2 1

$h(k) = k \% 7$

probes = 9

$LF = \frac{7}{7} = 1$



Load factor → Avg length of LL

No concept of 2/3 FULL.